M335 L EB

THE OPEN UNIVERSITY
Mathematics: A Third Level Course
M335 Studies in Pure Mathematics

STUDIES IN PURE MATHEMATICS

LOGIC

EXERCISE BOOKLET

Copyright © 1982 The Open University

SUP 08864 5

1.4

EXERCISES

Unit 3

Section 3.1

3.1 Let h be the function

$$\mathrm{Pr}[\mathrm{Cn}[\mathrm{sum}, \mathrm{id}_1^2, \mathrm{id}_2^2], \mathrm{Cn}[\mathrm{prod}, \mathrm{id}_4^4, \mathrm{Cn}[\mathrm{dif}, \mathrm{id}_1^4, \mathrm{id}_3^4]]].$$

Calculate the values of:
(i) $h(3, 1, 0)$;
(ii) $h(3, 1, 3)$.

In Exercises 3.2, 3.3 and 3.4, you may use any of the results about primitive recursive functions in Section 3.1 or in the Logic Handbook.

3.2 Show that the function

$$h(x) = \begin{cases} x - 2, & \text{if } x \geq 2, \\ 0, & \text{otherwise,} \end{cases}$$

is primitive recursive.

3.3 Given integers x and y, the Euclidean algorithm gives integers q and r such that

$$x = yq + r \quad \text{where } 0 \leq r < y.$$

q is called the quotient and r the remainder of x divided by y. Assuming that the function quot, defined by quot (x, y) = quotient of x divided by y, is primitive recursive, show that the following functions are primitive recursive:

(i) the function rem, defined by

$$\mathrm{rem}\,(x, y) = \text{remainder of } x \text{ divided by } y;$$

(ii) the function h, where

$$h(x, y) = {}^xC_y \left(= \frac{x!}{y!(x - y)!} \right),$$

with domain the set of all pairs (x, y) of natural numbers where $y \leq x$.

3.4 (i) Show that the function g, defined by

$$g(x) = \sum_{n<x} n^2,$$

is primitive recursive.

(ii) Show that the function l, where

$$l(x) = \begin{cases} x^2, & \text{if } x \text{ is odd,} \\ 0, & \text{if } x \text{ is even,} \end{cases}$$

is primitive recursive.

(iii) Show that the function h, where

$$h(x) = \sum_{\substack{n \text{ odd} \\ n<x}} n^2,$$

is primitive recursive. [Hint: Follow the technique of Problem 15, page 66 of *Unit 3*, using the function l in part (ii) instead of e.]

(iv) Show that the function k, where

$$k(x) = 1^2 + 3^2 + 5^2 + \ldots + (2x + 1)^2,$$

is primitive recursive.

Section 3.3

3.5 Let f be the function defined by

$$f(x, y, z) = (y \cdot e(x)) \div (z \cdot e(y)),$$

where e is the function of Problem 14, page 65 of *Unit 3*, defined by

$$e(x) = \begin{cases} 0, & \text{if } x \text{ is even,} \\ 1, & \text{if } x \text{ is odd.} \end{cases}$$

(i) Compute the values of

$\mathrm{Mn}[f](2,3)$, $\mathrm{Mn}[f](3,0)$, $\mathrm{Mn}[f](3,1)$, $\mathrm{Mn}[f](5,7)$, $\mathrm{Mn}[f](6,8)$.

(ii) Explain why $\mathrm{Mn}[f](1,2)$ is not defined.

(iii) For which pairs (x, y) of natural numbers is $\mathrm{Mn}[f](x, y)$ not defined?

(iv) What, in general, is the value of $\mathrm{Mn}[f](x, y)$ when it is defined?

Unit 4

Section 4.1

In the following exercises you need give only informal, as opposed to formal, definitions to show that functions are primitive recursive.

4.1 (i) Use the function d of Example 4, page 79 of *Unit 4*, to show that the condition 'x is even' on natural numbers x is primitive recursive.

(ii) Explain why the characteristic function c' of the condition 'x is odd' on natural numbers x cannot be defined by $c'(x) = d(x \div 1, 2)$.

(iii) Show that the condition 'x is odd' on natural numbers x is primitive recursive.

(iv) Show that the function f, defined by

$$f(x) = \begin{cases} x^2, & \text{if } x \text{ is even,} \\ 1, & \text{if } x \text{ is odd,} \end{cases}$$

is primitive recursive.

4.2 (i) Prove that the condition '$x^2 + y^2 = z$' on triples (x, y, z) of natural numbers is primitive recursive. (So, for example, $(2, 2, 8)$ satisfies the condition, as $2^2 + 2^2 = 8$, while $(8, 2, 2)$ does not, as $8^2 + 2^2 \neq 2^2$.)

(ii) Prove that the function h, defined by

$h(z)$ = the number of ways z can be expressed as the sum of two squares,

is primitive recursive.
(For example $25 = 0^2 + 5^2 = 5^2 + 0^2 = 3^2 + 4^2 = 4^2 + 3^2$, so $h(25) = 4$.)

(iii) Show that the condition 'z is the sum of two squares' on natural numbers z is primitive recursive.

4.3 (i) Show that the condition 'both x and y are divisible by z' on triples (x, y, z) of natural numbers is primitive recursive.

(ii) Show that the condition 'x and y are relatively prime' (i.e. 'x and y have no positive common factors other than 1') on pairs (x, y) of positive integers is primitive recursive. [Hint: How is the number of $z \leq y$ such that z divides both x and y related to whether x and y are relatively prime?]

(iii) Deduce that Euler's ϕ-function, with domain the set of positive integers and defined by

$\phi(x)$ = the number of positive integers $y \leq x$ such that x and y are relatively prime,

is primitive recursive.
(For instance, $\phi(1) = 1$, $\phi(2) = 1$, $\phi(3) = 2$, $\phi(4) = 2$. It may help to use these values of $\phi(x)$ to test and, if necessary, amend your solution.)

Unit 5

Section 5.3

5.1 Show that the formula

$((-\forall z\,(z = (y + x)\,\&\,y = \mathbf{0}) \rightarrow \exists x\, x = z) \vee (-\exists x\, x \doteq z \vee \forall z\,(z = (y + x)\,\&\,y = \mathbf{0})))$

takes truth value 1 under all interpretations of the symbols used in the formula.

Section 5.4

5.2 The following is a correct proof from which the assumption numbers have been omitted.

(1)	$(\phi \vee \psi)$	Ass
(2)	$-\phi$	Ass
(3)	$-\psi$	Ass
(4)	$(-\phi\,\&\,-\psi)$	Taut, (2), (3)
(5)	$-(-\phi\,\&\,-\psi)$	Taut, (1)
(6)	$((-\phi\,\&\,-\psi)\,\&\,-(-\phi\,\&\,-\psi))$	Taut, (4), (5)
(7)	$(-\psi \rightarrow ((-\phi\,\&\,-\psi)\,\&\,-(-\phi\,\&\,-\psi)))$	CP, (6)
(8)	ψ	Taut, (7)
(9)	$((\phi \vee \psi) \rightarrow \psi)$	CP, (8)

(i) What are the assumptions in force on each line?

(ii) What tautologies are being used on lines (4), (5) and (8)?

(iii) Show that each of the formulas in your answer to part (ii) is a tautology.

5.3 Find all the mistakes in the following proof. (When investigating a particular line, find the errors on it, if any, pretending that all the previous lines are correct.)

1	(1)	$(\phi \vee \psi)$	Taut
1	(2)	$(\phi\,\&\,\psi)$	Taut, (1)
2	(3)	ϕ	Taut, (2)
1	(4)	ψ	Taut, (1), (2)
	(5)	$(\psi \rightarrow (\phi \vee \psi))$	CP, (4)
1	(6)	$(\phi \vee \psi)$	Taut, (4), (5)

5.4 Write down a formal proof of $((\phi \rightarrow \psi) \rightarrow (-\psi \rightarrow -\phi))$ depending on no assumptions.

Unit 6

Section 6.3

6.1 Show that for all formulas ϕ, ψ, where the variable x does not occur free in ϕ,

$$\forall x\,(\phi \rightarrow \psi) \;\vdash\; (\phi \rightarrow \forall x\,\psi)$$

For which step of your argument do you need the condition that x does not occur free in ϕ?

Sections 6.5 and 6.6

6.2 The following correct (but needlessly elaborate!) formal proofs are incomplete because their assumption numbers are missing. Fill in these missing numbers.

(i)

(1)	$\forall x(\theta \to \psi)$	Ass
(2)	$(\theta \to \psi)$	UE, (1)
(3)	$\exists x\theta$	Ass
(4)	$(\theta \to (\theta \to \psi))$	Taut, (2)
(5)	θ	Ass
(6)	$(\theta \to \psi)$	Taut, (4), (5)
(7)	$\exists x(\theta \to \psi)$	EI, (6)
(8)	$\exists x(\theta \to \psi)$	EH, (7)
(9)	$(\exists x\theta \to \exists x(\theta \to \psi))$	CP, (8)

(ii)

(1)	$\forall x \forall y\, x = y$	Ass
(2)	$\forall y\, x = y$	UE, (1)
(3)	$x = x$	II
(4)	$x = y$	UE, (2)
(5)	$y = x$	SR, (3), (4)
(6)	$x = x$	UE, (2)

6.3 (i) What is wrong with the following attempt to show that, for all formulas ϕ,

$$\forall x - \phi \quad \vdash \quad - \exists x \phi \quad ?$$

1	(1)	$\forall x - \phi$	Ass
1	(2)	$-\phi$	UE, (1)
3	(3)	$\exists x \phi$	Ass
4	(4)	ϕ	Ass
1,4	(5)	$(\phi \,\&\, -\phi)$	Taut, (2), (4)
1,3	(6)	$(\phi \,\&\, -\phi)$	EH, (5)
1	(7)	$(\exists x \phi \to (\phi \,\&\, -\phi))$	CP, (6)
1	(8)	$-\exists x \phi$	Taut, (7)

(ii) Show that, for all formulas ϕ,

$$\forall x - \phi \quad \vdash \quad - \exists x \phi.$$

Unit 7

Section 7.2

7.1 Show that $\vdash_Q \exists y \forall x((x \cdot y) \cdot y) = \mathbf{0}$.

Section 7.3

7.2 Show that each of the following sentences is true in the interpretation N^* of Section 7.3 of *Unit 7*, and false in the interpretation N^{**}.

(i) $\forall x \forall y \forall z\,((x + y) \cdot z) = ((x \cdot z) + (y \cdot z))$

(ii) $\exists x\,(x = x' \,\&\, \forall y(-y = x \to -y = y'))$

7.3 For each sentence below, decide whether or not the sentence is a theorem of Q. If it is, give a formal proof that shows it is a theorem; if it is not a theorem, explain why not.

(i) $\forall x\,(x \cdot (\mathbf{0}' + \mathbf{0})) = ((x \cdot \mathbf{0}') + (x \cdot \mathbf{0}))$

(ii) $\forall x\,((\mathbf{0}' + \mathbf{0}) \cdot x) = ((\mathbf{0}' \cdot x) + (\mathbf{0} \cdot x))$

SOLUTIONS

3.1 The outermost operation used to define h is Pr. So h is of the form $\mathrm{Pr}[f, g]$, where the functions used in this primitive recursion construction are

$$f = \mathrm{Cn}[\mathrm{sum}, \mathrm{id}_1^2, \mathrm{id}_2^2]$$

and $$g = \mathrm{Cn}[\mathrm{prod}, \mathrm{id}_4^4, \mathrm{Cn}[\mathrm{dif}, \mathrm{id}_1^4, \mathrm{id}_3^4]].$$

As f is a function of two arguments and g is a function of four arguments, h must be a function of three arguments. Decomposing the definitions of f and g, we obtain

$$\begin{aligned} h(x, y, 0) &= f(x, y) = x + y, \\ h(x, y, s(t)) &= g(x, y, t, h(x, y, t)) \\ &= h(x, y, t) \cdot (x \dot{-} t). \end{aligned}$$

(We have used the variable t rather than the more usual z to avoid confusion with the zero function!)

(i) $h(3, 1, 0) = f(3, 1) = 3 + 1 = 4.$

(ii) To calculate $h(3, 1, 3)$ we need to calculate successively $h(3, 1, 0)$, $h(3, 1, 1)$, $h(3, 1, 2)$, and finally $h(3, 1, 3)$.

$h(3, 1, 0) = 4$ from part (i).

$$\begin{aligned} h(3, 1, 1) &= h(3, 1, s(0)) \\ &= g(3, 1, 0, h\ (3, 1, 0)) \\ &= 4 \cdot (3 \dot{-} 0) \\ &= 12 \end{aligned}$$

$$\begin{aligned} h(3, 1, 2) &= h(3, 1, s(1)) \\ &= g(3, 1, 1, h(3, 1, 1)) \\ &= 12 \cdot (3 \dot{-} 1) \\ &= 24 \end{aligned}$$

$$\begin{aligned} h(3, 1, 3) &= h(3, 1, s(2)) \\ &= g(3, 1, 2, h(3, 1, 2)) \\ &= 24 \cdot (3 \dot{-} 2) \\ &= 24 \end{aligned}$$

3.2 There are many ways of doing this, for instance the following.

The function pred is primitive recursive (Example 6, page 64 of *Unit 3*). Then $h(x) = \mathrm{pred}(\mathrm{pred}(x))$ for all x, so h is obtained by composition of primitive recursive functions and is thus primitive recursive. (This could also be expressed formally as $h = \mathrm{Cn}[\mathrm{pred}, \mathrm{pred}]$.)

Or, using pred again, and the method described on **BJ** page 77, lines -2 and -1, for defining functions of one variable by primitive recursion,

$$\begin{aligned} h(0) &= 0 \\ h(s(y)) &= g(y, h(y)) \end{aligned}$$

where $$\begin{aligned} g(y, t) &= \mathrm{pred}(y) \\ &\quad (= \mathrm{Cn}[\mathrm{pred}, \mathrm{id}_1^2](y, t)). \end{aligned}$$

As pred is primitive recursive, g and thus h are both primitive recursive.

Or, using the primitive recursive function dif of Example 8, page 64 of *Unit 3*,

$$\begin{aligned} h(x) &= \mathrm{dif}(x, 2) \\ &= \mathrm{dif}(x, s(s(z(x)))) \end{aligned}$$

so h is obtained by composing the primitive recursive functions dif, s and z and hence is primitive recursive. (This could also be expressed in formal notation as $h = \mathrm{Cn}[\mathrm{dif}, \mathrm{id}, \mathrm{Cn}[s, \mathrm{Cn}[s, z]]]$.)

3.3 (i) Given x and y,

$$\begin{aligned}\text{rem}(x,y) &= x - y \cdot \text{quot}(x,y)\\ &= \text{id}_1^2(x,y) \dot{-} \text{prod}(\text{id}_2^2(x,y), \text{quot}(x,y)).\end{aligned}$$

(Note that $x \geq y \cdot \text{quot}(x,y)$ is always true, so the '$-$' in the first line above can be replaced directly by '$\dot{-}$'.)

As $\dot{-}$, quot, prod, etc., are primitive recursive functions, and rem is obtained by composition of these, rem is also primitive recursive. (In formal notation, rem = Cn[dif, id_1^2, Cn[prod, id_2^2, quot]].)

(ii) Using the function fac of Problem 8, page 65 of *Unit 3*,

$$\begin{aligned}h(x,y) &= \text{quot}(\text{fac}(x), \text{fac}(y) \cdot \text{fac}(x \dot{-} y))\\ &= \text{quot}(\text{fac}(\text{id}_1^2(x,y)), \text{prod}(\text{fac}(\text{id}_2^2(x,y)), \text{fac}(x \dot{-} y))).\end{aligned}$$

As h is obtained by the composition of primitive recursive functions, it is primitive recursive.

Note that for $y \leq x$, $\dfrac{x!}{y!(x-y)!}$ is actually an integer, so the use of the quotient function gives the right answer. As a bonus (certainly not asked for by the question), the above function gives a sensible value for $h(x,y)$ when $y > x$. What is it?!

3.4 (i) Using the result of Problem 11, page 65 of *Unit 3*, with the function f defined by

$$f(y) = \text{prod}(y,y) \quad (= \text{Cn}[\text{prod}, \text{id}, \text{id}](y)),$$

so that f is primitive recursive, the function g is primitive recursive.

(ii) Using the function e of Problem 14, page 65 of *Unit 3*, and the function f above,

$$l(x) = \text{prod}(e(x), f(x)) \quad (= \text{Cn}[\text{prod}, e, f](x))$$

which is the composition of primitive recursive functions, and hence primitive recursive.

(iii) Using the technique of Problem 15, page 66 of *Unit 3*,

$$h(x) = \sum_{n<x} l(x) \qquad \text{where } l \text{ is as in part (ii).}$$

As l is primitive recursive, h is primitive recursive by the result of Problem 11, page 65 of *Unit 3*.

(iv) One way is to notice that $k(x)$ equals $h(2x+2)$, where h is as in part (iii). So letting j be the function Cn[sum, s, s], so that $j(x) = 2x + 2$, $k = \text{Cn}[h, j]$, so k is primitive recursive.

3.5 (i) $f(2,3,z) = (3 \cdot 0) \dot{-} (z \cdot 1) = 0$ for all z,

so $\text{Mn}[f](2,3) = 0$.

$f(3,0,z) = (0 \cdot 1) \dot{-} (z \cdot 0) = 0$ for all z,

so $\text{Mn}[f](3,0) = 0$.

$f(3,1,z) = (1 \cdot 1) \dot{-} (z \cdot 1)$.

The smallest value of z for which $f(3,1,z) = 0$ is $z = 1$, so $\text{Mn}[f](3,1) = 1$.

$f(5,7,z) = (7 \cdot 1) \dot{-} (z \cdot 1)$.

The smallest value of z for which $f(5,7,z) = 0$ is $z = 7$, so $\text{Mn}[f](5,7) = 7$.

$f(6,8,z) = (8 \cdot 0) \dot{-} (z \cdot 0) = 0$ for all z,

so $\text{Mn}[f](6,8) = 0$.

(ii) $f(1,2,z) = (2 \cdot e(1)) \dot{-} (z \cdot e(2))$
$= (2 \cdot 1) \dot{-} (z \cdot 0)$
$= 2$ for all z.

So for no z can $f(1,2,z)$ equal zero, hence $\mathrm{Mn}[f](1,2)$ is undefined.

(iii) $\mathrm{Mn}[f](x,y)$ is undefined precisely when

$$y \cdot e(x) > z \cdot e(y) \quad \text{for all } z.$$

If $e(y) = 1$ (so y is odd), this can never happen, so $\mathrm{Mn}[f](x, y)$ is undefined when $e(y) = 0$, i.e. y is even, and $y \cdot e(x) > 0$, i.e. y is positive and x is odd.

Thus $\mathrm{Mn}[f](x,y)$ is undefined for those pairs (x,y) where x is odd and y is even and positive.

(iv) If x is even,

$$f(x,y,z) = (y \cdot 0) \dot{-} (z \cdot e(y)) = 0 \quad \text{for all } z \text{ and } y,$$

so $\mathrm{Mn}[f](x,y) = 0$.

If x is odd and y is odd,

$$f(x,y,z) = (y \cdot 1) \dot{-} (z \cdot 1),$$

so the least value of z for which $f(x,y,z) = 0$ is $z = y$,
hence $\mathrm{Mn}[f](x,y) = y$.

If x is odd and $y = 0$,

$$f(x,y,z) = (0 \cdot 1) \dot{-} (z \cdot 0) = 0 \quad \text{for all } z,$$

so $\mathrm{Mn}[f](x,0) = 0$.

To summarize:

$$\mathrm{Mn}[f](x,y) = \begin{cases} 0, & \text{if } x \text{ is even, or if } y = 0, \\ y, & \text{if both } x \text{ and } y \text{ are odd,} \\ \text{undefined}, & \text{if } x \text{ is odd and } y \text{ is even and positive.} \end{cases}$$

4.1 (i) We need to show that the function c where

$$c(x) = \begin{cases} 1, & \text{if } x \text{ is even,} \\ 0, & \text{if } x \text{ is not even (i.e. odd),} \end{cases}$$

is primitive recursive.

Now x is even if and only if x is divisible by 2, i.e. $d(x,2) = 1$.

So it would seem that we can define c by $c(x) = d(x,2)$, but we must check that $c(x)$ takes the value 0 when x is not even.

Fortunately, $d(x,2) = 0$ if x is not divisible by 2, i.e. if x is not even, so this definition of c will do. As d is primitive recursive, so is c (using an analogue of Problem 9(i), page 65 of *Unit 3*). Hence the condition 'x is even' has a primitive recursive characteristic function, and is thus primitive recursive. (c is essentially the function d' of Solution 9, page 81 of *Unit 4*.)

(ii) Defining c' by $c'(x) = d(x \dot{-} 1, 2)$ almost works, as

$$c'(x) = \begin{cases} 1, & \text{if } x \text{ is odd,} \\ 0, & \text{if } x \text{ is even,} \end{cases}$$

for all x except $x = 0$, when $d(0 \dot{-} 1, 2) = d(0,2) = 1$, while as 0 is even we require $c'(0) = 0$. As this definition of c' fails for some value of x, we do not have a correct definition of the characteristic function of the condition.

(iii) We require a *correct* definition of c' in part (ii) in terms of known primitive recursive functions! There are many ways of doing this, for instance the following.

If c is the primitive recursive function obtained in the solution to part (i), then

$$\overline{\text{sg}}(c(x)) = \begin{cases} 0, & \text{if } x \text{ is even,} \\ 1, & \text{if } x \text{ is odd,} \end{cases}$$

so define c' by $c'(x) = \overline{\text{sg}}\ (c(x))$.

Or, if d is the primitive recursive function of Example 4, page 79 of *Unit 4*,

$$d(x+1, 2) = \begin{cases} 1, & \text{if } x \text{ is odd} \\ 0, & \text{if } x \text{ is even.} \end{cases}$$

So define c' by $c'(x) = d(s(x), 2)$, and use an analogue of Problem 9(i), page 65 of *Unit 3*.

Or, define $c'(x) = e(x)$, where e is the primitive recursive function of Problem 14, page 65 of *Unit 3*.

In any of these cases, c' is obtained by composing known primitive recursive functions and is itself primitive recursive. So the condition has a primitive recursive characteristic function and is thus primitive recursive.

(iv) There are a number of ways of doing this. Firstly one can recognize the definition of $f(x)$ as being of the form

$$f(x) = \begin{cases} g_1(x), & \text{if } C_1, \\ g_2(x), & \text{if } C_2, \end{cases}$$

where $g_1(x) = x^2 = \text{prod}(x, x)$
$g_2(x) = 1 = s(z(x))$,
so g_1 and g_2 are primitive recursive,

and C_1 is the condition 'x is even',
C_2 is the condition 'x is odd',
so that C_1 and C_2 are mutually exclusive, collectively exhaustive, and primitive recursive (by parts (i) and (iii)) conditions.

Thus, by definition by cases, f is primitive recursive.

Or this can be done by recognizing that f can also be defined by

$$f(x) = \begin{cases} x^2, & \text{if } x \text{ is even,} \\ 1, & \text{otherwise,} \end{cases}$$

and using Problem 5, page 78 of *Unit 4*, with $n = 1$, plus a similar discussion to the above.

Or, one can write down a formula for $f(x)$ similar to the one in the box on line 16 of **BJ** page 85, e.g.

$$f(x) = \text{prod}(x, x)\ c(x) + s(z(x))\ c'(x)$$

where c and c' are the primitive recursive functions of parts (i) and (iii), and deduce that f is primitive recursive.

4.2 (i) Using the primitive recursive function Eq of Example 1, page 77 of *Unit 4*, define the function c of three arguments by

$$c(x, y, z) = \text{Eq}(\text{sum}(\text{prod}(x, x), \text{prod}(y, y)), z)$$

so that $c(x, y, z) = 1$, if $x^2 + y^2 = z$
and $c(x, y, z) = 0$, if $x^2 + y^2 \neq z$.

Thus c is the characteristic function of the condition. Also, c is obtained by composition of known primitive recursive functions and is thus primitive recursive. Hence the condition is primitive recursive.

(ii) Essentially, the value of $h(z)$ is given by the sum of the values of $c(x, y, z)$ for appropriate values of x and y. As we cannot sum $c(x, y, z)$ for infinitely many x and y, we note that in no circumstances can $x^2 + y^2 = z$ for values of x or y greater than z. So it is enough to sum for values of x and y with $x \leq z$ and $y \leq z$. (Smaller bounds on x and y will do in general, but these bounds are needed for $z = 0$ or 1.)

In order to use this definition of h to show that h is primitive recursive, we use Example 3, page 79 of *Unit 4*. In fact, we use it twice. First, since the function c is primitive recursive (by part (i)), the function h' defined by

$$h'(x, z) = \sum_{y=0}^{z} c(x, y, z)$$

is primitive recursive, using the technique of Example 3. Now

$$h(z) = \sum_{x=0}^{z} h'(x, z),$$

so the same technique shows h is primitive recursive.

[Strictly speaking, the format of the technique of Example 3 requires that we should define h' by first defining a function g of three arguments by

$$g(x, t, z) = \sum_{y=0}^{z} c(x, y, t),$$

which is primitive recursive by Example 3, and then define h' by

$$h'(x, z) = g(x, z, z).$$

The use of the technique of Example 3 to define h from h' similarly needs an intermediate step in a formal argument. However, the sort of argument actually used in our solution of part (ii) is a perfectly acceptable *informal* argument.]

(iii) Clearly z is the sum of two squares if and only if $h(z) > 0$, so define the function c^* by

$$c^*(z) = \text{sg}(h(z))$$

so that

$c^*(z) = 1$, if $h(z) > 0$, i.e. if z is the sum of two squares,
$c^*(z) = 0$, if $h(z) = 0$, i.e if z is not the sum of two squares.

Thus c^* is the characteristic function of the condition. Also, c^* is obtained by composing primitive recursive functions, so it is a primitive recursive function, and the condition is primitive recursive.

4.3 (i) Using the function d of Example 4, page 79 of *Unit 4*, define the function c of three arguments by

$$c(x, y, z) = \text{prod}(d(x, z), d(y, z)),$$

so that $c(x, y, z) = 1$, if both $d(x, z) = 1$ and $d(y, z) = 1$,
i.e. if both x and y are divisible by z,
and $c(x, y, z) = 0$, otherwise.

Thus c is the characteristic function of the condition. As c is defined by composition of primitive recursive functions, c is also a primitive recursive function, and the condition is therefore primitive recursive.

(ii) One way of doing this is to count the number of z that divide x and y, which can be done by summing the values of $c(x, y, z)$ for all $z \leq y$ (as no divisor z of y can be greater than y). This sum will always be at least 1 as $c(x, y, 1) = 1$ (i.e. 1 divides both x and y for any positive x, y). However, this sum is greater than 1 if and only if x and y have a common divisor other than 1, i.e. x and y are *not* relatively prime.

Thus the function c^* defined by

$$c^*(x, y) = \overline{\text{sg}}\left(\left(\sum_{z=0}^{y} c(x, y, z)\right) \dot{-} 1\right)$$

is the characteristic function of the condition, as

$c^*(x, y) = 1$, if $\sum_{z=0}^{y} c(x, y, z) \leq 1$,
i.e. if only 1 divides both x and y,
and $c^*(x, y) = 0$, otherwise.

As c^* is obtained by composition of primitive recursive functions and by use of the technique of Example 3, page 79 of *Unit 4*, on a primitive recursive function, c^* is itself a primitive recursive function, and the condition is primitive recursive.

(iii) We should like to define ϕ by

$$\phi(x) = \sum_{y=0}^{x} c^*(x, y), \qquad (\dagger)$$

as the technique of Example 3, page 79 of *Unit 4*, could then be applied to show ϕ is primitive recursive. Unfortunately, the construction of c^* in our solution to part (ii), which gives $c^*(x, y)$ the desired value for positive integers x and y, also gives a value of 1 to $c^*(x, 0)$. Thus the right-hand-side of (†) actually equals $\phi(x) + 1$.

To allow for this, we shall define ϕ by

$$\phi(x) = \left(\sum_{y=0}^{x} c^*(x, y) \right) \dot{-} 1.$$

Thus ϕ can be obtained by general summation of the primitive recursive function c^* and composition with the primitive recursive difference function, and is therefore primitive recursive.

An attractive alternative approach would be to define ϕ by

$$\phi(x) = \sum_{y=1}^{x} c^*(x, y)$$

and show that the summation of the primitive recursive function $c^*(x, y)$ for values of y between 1 and x gives a primitive recursive function.

5.1 Let ϕ be the subformula $\forall z\,(z = (y + x)\,\&\,y = \mathbf{0})$ and let ψ be the subformula $\exists x\, x = z$. Then the formula becomes

$$((-\phi \to \psi) \vee (-\psi \vee \phi))$$

which has truth table

$((-\phi \to \psi)$		$(-\psi \vee \phi))$
0 1 1 1	1	0 1 1 1
0 1 1 0	1	1 0 1 1
1 0 1 1	1	0 1 0 0
1 0 0 0	1	1 0 1 0

Thus the formula takes truth value 1 under all interpretations of its symbols.

5.2 (i) 1 on line (1); 2 on line (2); 3 on line (3); 2, 3 on line (4); 1 on line (5); 1, 2, 3 on line (6); 1, 2 on line (7); 1, 2 on line (8); 2 on line (9).

(ii) On line (4): $((-\phi\,\&\,-\psi) \to (-\phi\,\&\,-\psi))$.
[The formula $((-\psi\,\&\,-\phi) \to (-\phi\,\&\,-\psi))$ would also have been correct.]

On line (5): $((\phi \vee \psi) \to -(-\phi\,\&\,-\psi))$.
On line (8): $((-\psi \to ((-\phi\,\&\,-\psi)\,\&\,-(-\phi\,\&\,-\psi))) \to \psi)$.

(iii) An answer using truth tables is required.

$((-\phi\,\&\,-\psi)$		$(-\phi\,\&\,-\psi))$
0 1 0 0 1	1	0 1 0 0 1
0 1 0 1 0	1	0 1 0 1 0
1 0 0 0 1	1	1 0 0 0 1
1 0 1 1 0	1	1 0 1 1 0

↑
tautology

$((\phi \vee \psi)$	$\rightarrow$	$-(-\phi \,\&\, -\psi))$
1 1 1	1	1 0 1 0 0 1
1 1 0	1	1 0 1 0 1 0
0 1 1	1	1 1 0 0 0 1
0 0 0	1	0 1 0 1 1 0

↑
tautology

| $((-\psi \rightarrow ((-\phi \,\&\, -\psi) \,\&\, -(-\phi \,\&\, -\psi))) \rightarrow \psi)$ | | |

6.1

1	(1)	$\forall x(\phi \rightarrow \psi)$	Ass
2	(2)	ϕ	Ass
1	(3)	$(\phi \rightarrow \psi)$	UE, (1)
1,2	(4)	ψ	Taut, (2), (3)
1,2	(5)	$\forall x \psi$	UI, (4)
1	(6)	$(\phi \rightarrow \forall x \psi)$	CP, (5)

The condition that x does not occur free in ϕ is needed to derive line (5), because the application of UI to the formula on line (4) requires that x does not occur free in any of the assumptions in force on line (4), and ϕ is one of these

6.2 (i) 1 on line (1); 1 on line (2); 3 on line (3); 1 on line (4); 5 on line (5); 1, 5 on line (6); 1, 5 on line (7); 1, 3 on line (8); 1 on line (9).

(ii) 1 on line (1); 1 on line (2); none on line (3); 1 on line (4); 1 on line (5); 1 on line (6).

6.3 (i) The use of EH on line (6) is incorrect, as the variable x may appear in the formula $(\phi \,\&\, -\phi)$ on line (5). (It would be a correct use if we were told, or knew, that x did not occur free in ϕ; but this is not the case here.)

(ii) One strategy would be as follows. Derive $(\phi \,\&\, -\phi)$ depending on the assumptions $\forall x -\phi$ and ϕ, as is done correctly in lines (1) to (5) of the proof in part (i). We should now like to derive a formula, depending on the same assumptions $\forall x -\phi$ and ϕ, of the form $(\theta \,\&\, -\theta)$ where the variable x does *not* occur free in θ, so that EH can legitimately be used to derive $(\theta \,\&\, -\theta)$ depending on the assumptions $\forall x -\phi$ and $\exists x \phi$. The technique of proof by contradiction now leads to the desired result. Fortunately, the formula $((\phi \,\&\, -\phi) \rightarrow (\theta \,\&\, -\theta))$ is a tautology for *any* formula θ, so we can pick any suitable formula θ and derive $(\theta \,\&\, -\theta)$ from $(\phi \,\&\, -\phi)$ as required: we do this on line (6) of the proof that follows, taking θ to be the formula $\forall x\, x = x$ to emphasize the artificiality of this trick!

1	(1)	$\forall x -\phi$	Ass
1	(2)	$-\phi$	UE, (1)
3	(3)	$\exists x \phi$	Ass
4	(4)	ϕ	Ass
1,4	(5)	$(\phi \,\&\, -\phi)$	Taut, (2), (4)
1,4	(6)	$(\forall x\, x = x \,\&\, -\forall x\, x = x)$	Taut, (5)
1,3	(7)	$(\forall x\, x = x \,\&\, -\forall x\, x = x)$	EH, (6)
1	(8)	$(\exists x \phi \rightarrow (\forall x\, x = x \,\&\, -\forall x\, x = x))$	CP, (7)
1	(9)	$-\exists x \phi$	Taut, (8)

As x does not occur free in $(\forall x\, x = x \,\&\, -\forall x\, x = x)$, nor in the assumptions other than ϕ in force on line (6), the use of EH on line (7) is now correct.

Another strategy would be to show that $\exists x \phi \vdash -\forall x -\phi$, as is done in Problem 2, page 57 of *Unit 6*, and add lines at the end to show that $\forall x -\phi \vdash -\exists x \phi$ as follows.

1	(1)	$\exists x \phi$	Ass	proof as in Solution 2, page 57 of *Unit 6*
.	.	.	.	
.	.	.	.	
.	.	.	.	
1	(8)	$-\forall x -\phi$	EH, (7)	
	(9)	$(\exists x \phi \rightarrow -\forall x -\phi)$	CP, (8)	
	(10)	$(\forall x -\phi \rightarrow -\exists x \phi)$	Taut, (9)	
11	(11)	$\forall x -\phi$	Ass	
11	(12)	$-\exists x \phi$	Taut, (11)	

Lines (9) to (12) illustrate an underlying general strategy for converting a proof of $A \vdash -B$, for some formulas A and B, into a proof of $B \vdash -A$.

7.1 The strategy is to derive the formula $((x \cdot \tau) \cdot \tau) = 0$ for some term τ that does not involve x (so that the final use of EI will be correct). Intuition suggests **0** as the sensible value of τ.

1	(1)	$\forall x\,(x \cdot \mathbf{0}) = \mathbf{0}$	Ass
1	(2)	$((x \cdot \mathbf{0}) \cdot \mathbf{0}) = \mathbf{0}$	UE, (1)
1	(3)	$\forall x\,((x \cdot \mathbf{0}) \cdot \mathbf{0}) = \mathbf{0}$	UI, (2)
1	(4)	$\exists y \forall x\,((x \cdot y) \cdot y) = \mathbf{0}$	EI, (3)

As assumption 1 is axiom *Q6* of *Q*, the sentence is a theorem of *Q*.

7.2 (i) In N^*, if all of x, y, z are interpreted as natural numbers, $(x + y) \cdot z$ equals the same natural number as $(x \cdot z) + (y \cdot z)$.

If at least one of x, y and z is interpreted as ω, there are two cases:

(a) if either $z = 0$ or both x and y equal 0, both $(x + y) \cdot z$ and $(x \cdot z) + (y \cdot z)$ equal 0;

(b) otherwise, both expressions equal ω.

In all cases, $(x + y) \cdot z = (x \cdot z) + (y \cdot z)$ for all values of x, y and z, so

$$\forall x \forall y \forall z\,((x + y) \cdot z) = ((x \cdot z) + (y \cdot z))$$

is true in N^*.

In N^{**}, interpret x as α, y as 0 and z as β. Then $(x + y) \cdot z$ is interpreted by $(\alpha + 0) \cdot \beta = \alpha \cdot \beta = \beta$, while $(x \cdot z) + (y \cdot z)$ is interpreted by $(\alpha \cdot \beta) + (0 \cdot \beta) = \beta + \beta = \alpha$, and $\alpha \neq \beta$. (Other choices of x, y, z will do.) So $((x + y) \cdot z) = ((x \cdot z) + (y \cdot z))$ does not hold for all values of x, y, z in N^{**}, so

$$\forall x \forall y \forall z\,((x + y) \cdot z) = ((x \cdot z) + (y \cdot z))$$

is false in N^{**}.

(ii) As the sentence starts with $\exists x$, to show it is true in N^* we must show that there is a value of x in N^* for which the rest of the sentence, i.e. $(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y'))$, is true in N^*. (Remember that $\exists x$ is interpreted as 'there is an x in the domain'.) Take $x = \omega$. (No other value of x will do in this example.) Then $\omega = \omega'$ and for all y in N^*, if $y \neq \omega$ (so that y is a natural number), then $y \neq y + 1 = y'$. So there is an x in N^* (namely ω) for which $(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y')$ is true in N^*.

To show the sentence is false in N^{***} we must show that there is no x in N^{***} for which $(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y'))$ is true. So we have to consider all possible values of x in N^{**} and demonstrate that in each case $(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y'))$ is false in N^{**}.

If x is a natural number, $x \neq x'$ so $x = x'$ is false, and, from the truth table of &, $(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y'))$ is false in N^{**}.

If $x = \alpha$, then $\alpha = \alpha'$; but for $y = \beta$ the formula $-y = x$ is true while, as $\beta = \beta'$, the formula $-y = y'$ is false. Thus $\forall y(-y = x \rightarrow -y = y')$ is false and $(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y')$ is false in N^{**} for this interpretation of x.

Similarly, if $x = \beta$, taking $y = \alpha$ shows that $(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y'))$ is false in N^{**}. So there is no x in N^{**} for which $(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y'))$ is true in N^{**}. Hence $\exists x\,(x = x' \,\&\, \forall y(-y = x \rightarrow -y = y'))$ is false in N^{**}.

7.3 (i) This is a theorem of Q. One formal proof is as follows.

1	(1)	$\forall x\,(x + \mathbf{0}) = x$	Ass
2	(2)	$\forall x\,(x \cdot \mathbf{0}) = \mathbf{0}$	Ass
	(3)	$(x \cdot (\mathbf{0}' + \mathbf{0})) = (x \cdot (\mathbf{0}' + \mathbf{0}))$	II
1	(4)	$(\mathbf{0}' + \mathbf{0}) = \mathbf{0}'$	UE, (1)
1	(5)	$(x \cdot (\mathbf{0}' + \mathbf{0})) = (x \cdot \mathbf{0}')$	SR, (3), (4)
	(6)	$((x \cdot \mathbf{0}') + (x \cdot \mathbf{0})) = ((x \cdot \mathbf{0}') + (x \cdot \mathbf{0}))$	II
2	(7)	$(x \cdot \mathbf{0}) = \mathbf{0}$	UE, (2)
2	(8)	$((x \cdot \mathbf{0}') + \mathbf{0}) = ((x \cdot \mathbf{0}') + (x \cdot \mathbf{0}))$	SR, (6), (7)
1	(9)	$((x \cdot \mathbf{0}') + \mathbf{0}) = (x \cdot \mathbf{0}')$	UE, (1)
1,2	(10)	$(x \cdot \mathbf{0}') = ((x \cdot \mathbf{0}') + (x \cdot \mathbf{0}))$	SR, (8) (9)
1,2	(11)	$(x \cdot (\mathbf{0}' + \mathbf{0})) = ((x \cdot \mathbf{0}') + (x \cdot \mathbf{0}))$	SR, (5), (10)
1,2	(12)	$\forall x\,(x \cdot (\mathbf{0}' + \mathbf{0})) = ((x \cdot \mathbf{0}') + (x \cdot \mathbf{0}))$	UI, (11)

As assumptions 1 and 2 are axioms *Q4* and *Q6* of Q, we have shown the sentence is a theorem of Q.

In our solution we have not bothered to use axiom *Q7* to reduce the term $(x \cdot \mathbf{0}')$ to $((x \cdot \mathbf{0}) + x)$ which then reduces, using axiom *Q4* , to $(\mathbf{0} + x)$, as this, while not incorrect, would serve only to lengthen the proof. If you used this method, we hope you did not claim that $(\mathbf{0} + x)$ equalled x, as in Problem 2(ii)(a), pages 75 and 76 of *Unit 7*, we showed that $\forall x\,(\mathbf{0} + x) = x$ is not a theorem of Q.

When drafting our solution to this exercise—before this booklet was printed—we made an interesting and tempting error in the use of the Substitution Rule, as follows.

		.	
		.	
		.	
...	(k)	$\tau_2 = \tau_1$	...
...	$(k + 1)$	$\tau_3 = \tau_1$	...
...	$(k + 2)$	$\tau_2 = \tau_3$	SR, (k), $(k + 1)$

This is an incorrect use of the rule, but the desired conclusion, namely $\tau_2 = \tau_3$, can be arrived at by inserting two new lines in between the old lines $(k + 1)$ and $(k + 2)$, as follows.

			.		
			.		
			.		
		...	(k)	$\tau_2 = \tau_1$	...
		...	$(k + 1)$	$\tau_3 = \tau_1$	...
new lines			$(k + 2)$	$\tau_3 = \tau_3$	II
		...	$(k + 3)$	$\tau_1 = \tau_3$	SR, $(k + 1)$, $(k + 2)$
old line	$(k + 2)$	...	$(k + 4)$	$\tau_2 = \tau_3$	SR, (k), $(k + 3)$

(ii) This is not a theorem of Q. In the interpretation N^{**} take $x = \alpha$.
Then $(0' + 0) \cdot x = (1 + 0) \cdot \alpha = 1 \cdot \alpha = \alpha$,
while $(0' \cdot x) + (0 \cdot x) = (1 \cdot \alpha) + (0 \cdot \alpha) = \alpha + \alpha = \beta \neq \alpha$.
Thus $((0' + 0) \cdot x) = ((0' \cdot x) + (0 \cdot x))$ is not true for all values of x in N^{**}.
Hence $\forall x((\mathbf{0}' + \mathbf{0}) \cdot x) = ((\mathbf{0}' \cdot x) + (\mathbf{0} \cdot x))$ is false in N^{**}.
All the axioms of Q are true in N^{**}, so by the Correctness Theorem, the sentence $\forall x((\mathbf{0}' + \mathbf{0}) \cdot x) = ((\mathbf{0}' \cdot x) + (\mathbf{0} \cdot x))$ is not a theorem of Q.

Printed in The United Kingdom
by The Open University